Docket No. **8886.001.00**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: **Dennis Wayne HURST et al.**

FILING DATE: **Concurrently Herewith**

FOR: **INTERNET SECURITY ANALYSIS SYSTEM AND PROCESS**

LIST OF INVENTORS' NAMES AND ADDRESSES

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

SIR:

Listed below are the names and addresses of the inventors for the above-identified patent application.

Dennis Wayne HURST
515 Old Providence Ct.
Alpharetta, GA 30004

Darrin Ray BARRALL
564 Cowan Rd.
Conyers, GA 30094

Caleb Ikaki SIMA
1103 Riverstone Place
Woodstock, GA 30188

A declaration containing all the necessary information will be submitted at a later date.

Respectfully Submitted,

LONG ALDRIDGE & NORMAN LLP

Kenneth D. Springer
Registration No. 39,843

Date: November 28, 2000

Sixth Floor
701 Pennsylvania Avenue, N.W.
Washington, D.C. 20004
Tel. (202) 624-1200
Fax. (202) 624-1298

Docket No. 8886.001.00
-- PATENT --

SPECIFICATION

TITLE OF THE INVENTION

INTERNET SECURITY ANALYSIS SYSTEM AND PROCESS

BACKGROUND OF THE INVENTION

1) Field of the Invention

This invention pertains to the field of computer security, and more particularly, to a system and process for analyzing potential security flaws in an Internet Web site.

2) Description of the Related Art

Increasingly, Internet Web sites are being exposed to external attack, or "hacking," which has been defined as the act of penetrating a closed computer system to gain access to knowledge and information that is contained within. Individuals attack Web sites for a variety of reasons. A report by the United States Federal Bureau of Investigation (FBI) indicates that the originators of proprietary information theft can be classified into six categories. The report shows 35% of the criminals were discontented employees, 28% hackers, 18% other U.S. companies, 11% foreign companies, 8% foreign governments, and 10% miscellaneous. Examples of some well-known attack methods include e-mail bombing, denial-of-service attacks, Trojan horses, worms, and simple back-door entry to a Web site. These attacks can not only cripple or shut down access to an attacked Web site, but can result in unauthorized access to confidential customer information, including access passwords, and even credit card account numbers. The resulting damage to a commercial Web site can easily run into the millions of dollars.

According to the most recent FBI report, cyber crimes increased from 547 in 1998 to 1,154 in 1999. The FBI and the Computer Security Institute (Silicon Valley) found that 62% of information security officials reported security breaches in 1999 (National Journal's Technology Daily). These break-ins resulted in $123 million losses from fraud, information theft, sabotage, and viruses.

Many companies use a proprietary base software product for conducting business and are focusing on tools for managing risk, and controlling sabotage against their applications. According to a Cyber-source study of online E-tailors, 75% consider fraud to be a problem, and 62% consider it to be a serious problem. The NIPC, FBI, and United States Treasury, along with the President, have committed themselves to working along side the private sector in putting these concerns to rest. Both the Economic Espionage Act of 1996, and the Theft of Trade Secrets Act (section 1832) have caused organizations to react promptly to damaging acts of violations.

Yet this problem is still not controlled and is in need of counter-measures. In January of 2000, several denial-of-service (DOS) attacks occurred to well known E-commerce Web sites such as YAHOO® and E-BAY®. Incidents like this have brought the issue of Web site security directly into the public limelight. Of all the individuals polled on the questions of online banking, 65% stated that security was the main concern.

Maintaining control of electronic fraud can be a time consuming process for E-commerce companies, banks, brokerage firms, and electronic billing/payment providers. Both the network administrators and Web-masters do not have the proper tools to detect Web-based vulnerabilities. The complexity of information, separate system options, assessing the significance of penetrations, and a decision for correction is not prevalent in today's workplace. On-line fraud has a special significance for an E-commerce site. The fear of security exploits can cause a negative impact on consumer confidence in an E-commerce site, which ultimately destroys the brand's image.

To prevent such attacks, Web site administrators manually search for and close potential security holes in their own Web sites. Because most Web sites undergo changes over time, and because new vulnerabilities and attack techniques are continually being developed, the Web site administrators must continually probe their sites for security weaknesses. This is time-consuming and fraught with the likelihood of undetected security flaws.

As the Internet continues to expand, more and more Web sites are being developed and operated by less experienced and trained personnel. And it becomes more and more difficult for all of these individuals to be knowledgeable in all of the latest techniques for hacking a Web site. This increases the potential for Web site security flaws to exist which can be exploited by hackers.

Accordingly, there is a need for an advanced Web security analysis system and process that can be used by Web site developers and administrators to identify security flaws in their Internet Web sites. It would also be advantageous to provide such a system and process which can be used by third party individuals who lack specific knowledge of an individual Web site's architecture and design. It would be further advantageous to provide such a system and process which is automated, and which performs a security check without significant manual user intervention. Other and further objects and advantages will appear hereinafter.

SUMMARY OF THE INVENTION

The present invention comprises a system and process for analyzing potential security flaws in an Internet Web site.

In one aspect of the invention, an Internet security analysis system and process checks a target Internet Web site against a predetermined set of known exploits.

In another aspect of the invention, an Internet security analysis process is automated to execute without significant manual user interaction.

In yet another aspect of the invention, an Internet security analysis process is recursive, gathering information of security vulnerabilities and then exploiting that information to search for additional security vulnerabilities.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a functional block diagram

Figure 2 is a flowchart of a method of parsing through a Web site to identify possible security holes;

DESCRIPTION OF THE PREFERRED EMBODIMENTS

In a preferred embodiment, an Internet security analysis process is initiated by a Web site administrator for a target Web site to be analyzed. The process may be executed by an Internet security analysis system, which may be a personal computer having Internet access to the target Web site. Alternatively, a user may conduct the Internet security process through an Internet security analysis Web site. In that case, the Internet analysis system may be accessible via a Web server hosting the Internet security analysis Web site. Other embodiments are possible.

Figure 1 is a block diagram of a preferred embodiment of an Internet security analysis system 100 for performing a security analysis process for a target Web site. In a preferred embodiment, the Internet security analysis system 100 includes at least one central processing unit (CPU) 105 having associated therewith memory 110 and a mass data storage device 115, and optionally a display device 120, a data input device 125 (e.g., keyboard; mouse), and a network connection device 130, all connected to each other via a communication bus 135. The data storage device 115 includes nonvolatile data storage means, such magnetic disk drive units, optical disk drive units, removable disk drive units, tape media, or any combination thereof. The network connection device 130 includes hardware and software for establishing a data link connection between the Internet security analysis system 100 and a target Web site via the Internet. Preferably, the network connection device 130 includes circuitry to tie directly to the Internet via T1, T4 or similar high bandwidth data lines as would be understood by one skilled in the art.

The Internet security analysis system 100 may include two or more integrated computer units, each having a separate CPU 105 having associated therewith memory 110 and/or a mass data storage device 115, and optionally a display device 120 and a data input device (e.g., keyboard; mouse) 125. The communication bus 135 may include two or more internal buses for integrated computer units, together with an external bus connecting two or more integrated computer units.

In a preferred embodiment, the Internet security analysis system 100 executes one or more software routines to perform an automatic security analysis process. The automatic security analysis process analyzes a target Web site to identify security flaws. Preferably, the security analysis process executes a method of parsing through a target Web site to search for common security flaws. For example, in a preferred embodiment, the method includes analyzing the hyper-text markup language (HTML) script for the Web pages of the Web site and searching for commented-out references. Such commented-out references may include uniform resource locators (URLS) of Internet addresses for Web pages or other objects which are not intended to be accessed by a visitor to the Web site. Such Web pages or objects may contain data resident on a server hosting the target Web site which is not intended to be available to a visitor. As another example, in a preferred embodiment, the method includes bombarding the data entry fields of any Web forms of the target Web site with data in an attempt to break the target Web site.

A flowchart for a preferred embodiment process 200 of automatically parsing through a target Web site to identify possible security holes is shown in Figure 2. The process 200 may be initiated by a user and executed by a software program running on the Internet security analysis system 100, which as described above may be preferably a standalone user computer, or a computer accessible to a user via the Internet, etc. Typically, the process 200 is initiated by an administrator of the target Web site.

In a first step 205, an Internet security analysis computer establishes an Internet connection with a Web server hosting an Internet Web site for which a security inspection will be performed (the "target Web site"). The Internet security analysis computer retrieves a default Web page for the target Web site, e.g., "Index.html."

In a step 210, the Internet security analysis system 100 parses through the script (e.g., HTML code) for the default Web page to search for any linked-to Web pages or other objects which are referenced. The URL or address of each linked-to Web page is then stored in a database at the Internet security analysis computer. In particular, the Internet security analysis computer parses through Web pages looking for anchor links and JAVASCRIPT® links.

In a step 215, each linked-to Web page is then in turn retrieved by the Internet security analysis computer and the script for the retrieved Web page is parsed to search for any further linked-to Web pages. This process is repeated for each link and each linked-to Web page that is found. Preferably, the "depth" of linked-to Web pages which are retrieved may be set by a user to any value, so that only first or second level linked-to Web pages may be retrieved. Alternatively, only linked-to Web pages which are hosted on a same server, or which have a same domain address, as the default Web page may be retrieved.

As a result of the steps 205-215, the Internet security analysis system builds a valid Web page database with an entire map of the valid Web pages of the target Web site.

Then, in a step 220, the default Web page for the target Web site is parsed for any hidden Uniform Resource Locators (URLs), such as commented-out code, or any reference to a file or Web page that is not linked-to (or HREF'ed). These hidden URLs are then checked against the valid Web page database to insure that the Internet security analysis system has not already retrieved the corresponding hidden Web page or object. If the corresponding hidden Web page or object has not previously been retrieved, then the address is stored in a prioritized database of "vulnerable" Web pages at the Internet security analysis computer.

In a step 225, each hidden Web page or object is then in turn retrieved by the Internet security analysis system and the script is parsed to search for any further hidden URL's. If any further hidden URL's are found, then the Internet security analysis system retrieves the corresponding hidden Web page or object for the hidden URL and parses through it to search for any still further hidden URLs. This process is repeated for each link and each linked-to Web page or object that is found.

As a result of the steps 220-225, the Internet security analysis system builds a vulnerable Web page database with an entire map of the “vulnerable” Web pages of the target Web site. Together, the steps 205-225 constitute a “Webcrawl” of the entire target Web site, performed by parsing through each retrieved Web page in turn to find additional links.

Preferably, when the Internet security analysis system 100 parses through each retrieved Web page, it performs a keyword search to detect points of interest. For instance, if a Web page is retrieved thru the Webcrawl with the word "admin" located in the Web page, the Internet security analysis system 100 will flag it as a "point of interest" and name it as "possible administration Web page."

Then, in a step 230, the Internet security analysis system 100 checks a predetermined - but extensible - list of known, common security vulnerabilities. Typically, these security vulnerabilities are “exploits” which have become well known to security experts. The Internet security analysis system scans the target Web site to determine whether or not any of the exploits are present at the target Web site.

For example, assume a case where the following three security vulnerabilities are to be checked:

```
/security/exploit.asp
/security/vuln.asp
/security/b00m.asp
```

In that case, the Internet security analysis system will issue a request to the target Web site for the directory "security." The Web server for the target Web site will issue either a true or false response, depending on whether the directory exists or not. If the response is true (indicating that the directory does exist), then the above-mentioned three security vulnerabilities are checked to determine whether they exist.

In the above case, it is also possible to perform a keyword search on each response received to detect points of interest. For instance, if a response is received with the keyword "<DIR>" included, the Internet security analysis system 100 will flag it as a vulnerability.

As another example, assume that the “Webcrawl” in the steps 205-225 retrieved the URL "/scripts/db/msadcs.dll" and stored it in the vulnerable Web pages database, and that "/msadc/msadcs.dll" is stored in the list of known, common security vulnerabilities. In that case, in the step 230, the Internet security analysis system 100 takes the actual filename of the check (msadcs.dll) and flags it as being present in "/scripts/db/msadcs.dll." This allows the Internet security analysis system 100 to find a security vulnerability even though it is out of its normal location. Once a security vulnerability is found in an alternate directory the Internet security analysis system 100 will check for other files that typically reside with the file that was found. For example, if “msadcs.dll” is usually in a folder that also contains “msadcs2.dll”, then the Internet security analysis system 100 would also check for the existence of “msadcs2.dll” in the “/scripts/db” folder where “msadcs.dll” was found.

Another security vulnerability check which may be employed by the Internet security analysis system 100 is common file name checking. For example, the Internet security analysis system 100 will recursively try to request WS_FTP.LOG from all directories retrieved from the Webcrawl.

Yet another security vulnerability check which may be employed by the Internet security analysis system 100 is port scanning (scanning a machine for open ports) integrated with Web server discovery. By taking the information retrieved from a port scanner (port open on any given host) the Internet security analysis system 100 will attempt to determine whether that port is supporting HTTP or HTTPS. If the port returns valid on any HTTP or HTTPS request then the server is marked as a Web server sitting on that specific port. The Internet security analysis system 100 will go through an entire range of hosts, trying each port with this method. When completed, the Internet security analysis system 100 will have a list of hosts that are currently hosting a Web server and the port or ports on which the Web server is listening. The Internet security analysis system 100 will then take this list and pass it to the "Webcrawl engine" portion of the Internet security analysis system 100. This allows an easy ability to scan an entire network and look for Web vulnerabilities

If a security vulnerability exists, then in a step 235 a record of the security vulnerability is added to a security vulnerability database maintained at the Internet security analysis system 100.

Accordingly, an efficient and intelligent vulnerability search is conducted. Because a Web site may be very large and contain hundreds or more Web pages and dozens or more exploits must be checked, the above-described process of determining the appropriateness of a security check before actually issuing the check greatly expedites the automated security check process.

Advantageously, in a step 240 the Internet security analysis system further exploits each of the vulnerabilities detected in the step 235 and added to the security vulnerability database to search for further security vulnerabilities and to gather other information regarding the target Web site. This process is recursively repeated until no new data is obtained.

In a step 245, the Internet security analysis system applies a predetermined - but extensible - list of hack methods to the data which was retrieved in the steps 205-240 to identify security vulnerabilities.

In a preferred embodiment, a first hack method comprises the Internet security analysis system issuing a series of requests, such as:

```
GET/$SERVER_NAME/$SCRIPT_NAME?$VAR1=/../../../boot.ini
GET/$SERVER_NAME/$SCRIPT_NAME?$VAR2=/../../../boot.ini
.
.
.
```

These requests check every script name and variable at the target Web site for a "Directory traversal exploit." If any of these requests return a positive result, then a security vulnerability has been found.

In a second hack method, the Internet security analysis system searches for buffer overflows by issuing a series of requests such as:

```
GET/$SERVER_NAME/$SCRIPT_NAME?$VAR1=AAAAAAA . . . (2048 more
A'S)
GET/$SERVER_NAME/$SCRIPT_NAME?$VAR2=AAAAAAA . . . (2048 more
A'S)
.
.
.
```

These requests check every script name and variable at the target Web site for unexpected results in the case of a buffer overflow. If any unexpected data is returned, then a security vulnerability has been found.

In a third hack method, the Internet security analysis system searches for old files or variables which are still present on a Web server for the target Web site, bit which are not intended to be available to a visitor. The Internet security analysis system searches issues a series of requests such as:

```
GET/$SERVER_NAME/$SCRIPT_NAME.old
GET/$SERVER_NAME/$SCRIPT_NAME.bak
GET/$SERVER_NAME/$SCRIPT_NAME.bac
```

These requests are issued for every script name and variable at the target Web site.

Although a preferred embodiment process includes the three hack methods described above, additional hack methods are also possible and may be employed.

For example, where the Web site includes a Web form, intentionally invalid data may be entered into one or more data entry fields of the Web form and submitted to the Web site to determine if any invalid or unexpected results are produced by the Web site. This process may recursively submit various combinations of invalid data entered into the various data entry fields of the Web form. Once a security vulnerability is detected in this way, the data producing the vulnerability may then be used as a starting point for testing all other Web forms at the target Web site.

Preferably, a user may also employ cookie or HTML manipulation to search for security vulnerabilities. The Internet security analysis system 100 will allow a user to intercept the communication between a browser and a Web server and give the user the option to change or manipulate the data being sent between the client and server before the data is actually sent. This allows a user to manipulate the communication in any way, including cookies and over SSL communication. The Internet security analysis system 100 allows a user to create an HTTP or HTTPS request from either an existing request that has already been send to the server or to create a new request on the fly, including the ability to add, modify or not send any of the following: the URL; parameters that are sent to the server; any Header that is sent in the request; cookies that are sent to the server; and the method that is used to send the request (i.e.: HEAD, POST, GET).

If during the Webcrawl in the steps 205-225, the Internet security analysis system 100 encounters what it deems to be a login page, it will give the option to the user to "brute force" the login (try multiple logins and passwords until a successful one is retrieved), or to automate the login so that the Internet security analysis system 100 will brute force it automatically, and the user does not have to stop the scan and will brute force it automatically.

Preferably, there are two methods of brute forcing a login screen it can either be a form login or a basic authorization "pop-up box" login.

The Internet security analysis system 100 also preferably supports passing of user defined user names and passwords that are sent to the target server. This facilitates testing of target Web sites that require a login name and password to gain access to the target Web site, or a folder on the target Web site.

Moreover, in some cases one or more of the above-described hack methods may not be used. Optionally, the user may choose the hack methods which are employed.

If any of security vulnerabilities have been found in the step 245, then in a step 250 the Internet security analysis system adds a record for the security vulnerability to the security vulnerability database.

Finally, in a step 255, the Internet security analysis system prioritizes all security vulnerabilities which have been identified in the steps 205-250, and presents the prioritized list to the user, for example on a computer display screen. For example, it is known that a "SQL server error" is high risk vulnerability, and accordingly, this will be communicated to the user via the prioritization. Preferably, the Internet security analysis system also suggests how the user may eliminate the detected security vulnerabilities.

In a preferred embodiment, a user may disable the automatic operation and instead control one or more aspects of how the Internet security analysis process executes. For example, the user may be allowed to change the values sent to the target Web site in the step 245. The user may also view any malformed "cookies" which are returned during the Internet security analysis process. The user may manually edit and change the HTML code for any Web page which is retrieved from the Web site and then submit the changed Web page back to the server for the target Web site. This facilitates scanning of applications where specific user interaction (e.g., username and password) are required to gain access to the application, or where an automated scan could be dangerous to the application. One example of an application where an automated scan could be dangerous would be administrative application that has elevated access to the target Web site.

While preferred embodiments are disclosed herein, many variations are possible which remain within the concept and scope of the invention. Such variations would become clear to one of ordinary skill in the art after inspection of the specification, drawings and claims herein. The invention therefore is not to be restricted except within the spirit and scope of the appended claims.